

certified digital certificate hosting services (z/OS V1R5 with RACF).

Centralized key management on z/OS, with its management, access controls, and auditability, has been used in production for over a decade.

Watch this space

In the future, IBM intends to offer outboard encryption for devices in the TotalStorage family, and continue leveraging the key management functions provided by ICSF. Of course, we intend to keep extending our key management facilities, too.

After all, why would we start over?

For more information

See the statement of direction in “IBM System z9 – the server built to protect and grow with your on demand enterprise,” IBM United States Hardware Announcement 105-241, dated July 27, 2005.

A little-LDAP’ll-do-ya

BY TOM SIRC AND ASHWIN VENKATRAMAN

Consider this scenario: A test environment has three WebSphere Application Server for z/OS cells (a logical grouping of servers and nodes); lets call them T1, T2, and T3. All three cells use a Local operating system (OS) user registry, and each has its own user and group structure because different application development teams are testing their own Java 2 Enterprise Edition (J2EE) application security. An issue comes up – we need to figure out how to share a cell between different application development teams without sacrificing security.

The security limitation we describe above is that every application running in a cell must share the user registry. Each user that needs access to an application running on the cell needs to be added to the authorization list for the entire WebSphere Application Server space. This creates the burden of managing user security and violates a security practice because users who should not have access to an application are still given the access to connect to WebSphere Application Server because of another application’s needs. Having multiple WebSphere Application Server cells to provide security isolation is usually the way to deal with this problem.

We face other problems when developing an application that exploits form-based authentication on the T2 cell. At one point the T2 cell needs to be down but our development deadline can’t slip. And development and testing of our application has to be completed on the T3 cell even though we have a different set of users for that cell.

Our first thought is to give the developers of the application access to the T3 cell. This requires multiple user IDs and Enterprise Java Beans authorization role (EJBROLE) access changes in RACF. We also need to remember to undo these changes after the T2 cell back online. To solve this user management issue, we realized that a little-LDAP’ll-do-ya! We pointed our T3 cell’s user registry to a Lightweight Directory Access Protocol (LDAP) server that already had the user and group structure for the application. We are able to change this setting in a short amount of time because the environment already exists, and the change won’t leave negative side-effects after the cell is switched back to the Local OS user registry.

Using an LDAP authentication scheme in a WebSphere Application Server

environment is a simpler, more secure way of allowing multiple development teams to use the same WebSphere Application Server cell without having to provide unnecessary privileges in RACF. For this to happen, J2EE application developers need to add the user-to-role mappings to the enterprise archive (EAR) deployment descriptor because the RACF EJBROLE class no longer handles that function. To add the mappings, first understand the user and group structure in the LDAP directory. Only one application development team can use the WebSphere Application Server cell at any one time. If security is a concern in your environment and you can’t run multiple WebSphere Application Server cells, an LDAP user registry is the alternative to consider!

